

Exhibit B to the
Declaration of Imran A. Khaliq In Support
Of Visto's Opening Claim Construction
Brief Under P.R. 4-5(a)

United States Patent [19][11] **Patent Number:** **6,023,708****Mendez et al.**[45] **Date of Patent:** **Feb. 8, 2000**

[54] **SYSTEM AND METHOD FOR USING A GLOBAL TRANSLATOR TO SYNCHRONIZE WORKSPACE ELEMENTS ACROSS A NETWORK**

[75] Inventors: **Daniel J. Mendez**, Mountain View;
Mark D. Riggins, San Jose; **Prasad Wagle**, Santa Clara; **Christine C. Ying**, Foster City, all of Calif.

[73] Assignee: **Visto Corporation**, Mountain View, Calif.

[21] Appl. No.: **08/865,075**

[22] Filed: **May 29, 1997**

[51] **Int. Cl.**⁷ **G06F 17/30**

[52] **U.S. Cl.** **707/203; 707/4; 707/201; 707/10; 395/200.35**

[58] **Field of Search** **707/10, 203, 201, 707/100, 4; 395/200.36, 200.48, 200.35**

[56] **References Cited****U.S. PATENT DOCUMENTS**

4,831,582	5/1989	Miller et al.	707/104
4,897,781	1/1990	Chang	364/200
5,263,157	11/1993	Janis	707/9
5,386,564	1/1995	Shearer et al.	395/650
5,392,390	2/1995	Crozier	395/161
5,572,643	11/1996	Judson	395/793
5,581,749	12/1996	Hossain et al.	707/1
5,613,012	3/1997	Hoffman et al.	382/115
5,623,601	4/1997	Vu	395/187.01
5,627,658	5/1997	Connors et al.	358/407
5,634,053	5/1997	Noble et al.	707/4
5,647,002	7/1997	Brunson	380/49
5,652,884	7/1997	Palevich	395/651
5,666,553	9/1997	Crozier	395/803
5,680,542	10/1997	Mulchandani	395/183.04

(List continued on next page.)

OTHER PUBLICATIONS

Margaret J. Brown, "The Visto Briefcase Pro Puts Your PIM On The Internet", URL: <http://www.zdnet.com/zdnn/stories/>

zdnn_display/0,3440,341892,00.html, Aug. 13, 1998, 1 page.

Web site entitled "Bookmark Translator 2.0: This Utility transform Microsoft Internet Explore's bookmarks in the format valid for Netscape Navigator and viceversa," Enzo Marinacci, Rome—Jul. 1997, URL=<http://www.bns.it/emware/BookmarkTranslator-uk.htm>, pp. 1–4.

Article by Bellocin et al., entitled: "Network Firewalls" Published by IEEE Communications Magazine Sep. 1994, pp. 50–57.

Article by Steffen Stempel, entitled: "IPAccess—An Internet Service Access System for Firewall Installations" Published by IEEE Communications Magazine Feb. 16, 1995, pp. 31–41.

Article by Braun et al., entitled: "Web Traffic Characterization: an assessment of the impact of caching documents from NCSA's web server" Published by Elsevier Science B.V. 1995 pp. 37–51.

Article by Nelson et al., entitled: "Security for Infinite Networks" Published by IEEE Communications Magazine on Aug. 22, 1995, pp. 11–19.

(List continued on next page.)

Primary Examiner—Thomas G. Black

Assistant Examiner—Charles L. Rones

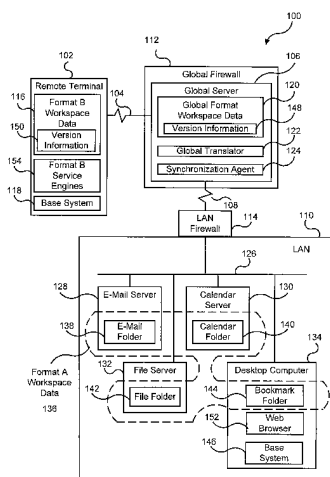
Attorney, Agent, or Firm—Graham & James LLP

[57]

ABSTRACT

A system uses a global translator to automatically synchronize multiple copies of a workspace element across different formats between multiple sites in a secure network environment, independent of whether the sites are protected by site firewalls. The secure network environment includes a global server connected to multiple clients. The system includes a first store for storing a first workspace element in a first format, a second store for storing a second workspace element which is an independently modifiable copy of the first workspace element in a second format, a communications channel coupling the first store to the second store, synchronization means for synchronizing the first workspace element and the second workspace element, and a translator for translating between the first format and the second format.

34 Claims, 7 Drawing Sheets



6,023,708

Page 2

U.S. PATENT DOCUMENTS

5,682,524	10/1997	Freund et al.	395/605	5,790,425	8/1998	Wagle	1/1
5,684,990	11/1997	Boothby	707/203	5,832,483	11/1998	Barker	707/8
5,701,423	12/1997	Crozier	395/335	5,870,759	2/1999	Bauer et al.	707/201
5,706,502	1/1998	Foley et al.	395/610	5,961,590	10/1999	Mendez et al.	709/206
5,713,019	1/1998	Keaten	395/610	OTHER PUBLICATIONS			
5,715,403	2/1998	Stefik	705/44	Article by Greenwald et al., entitled: "Designing an Academic Firewall: Policy, Practice, and Experience with SURF" Published by IEEE Communications Magazine on Feb. 22, 1996, pp. 79-92.			
5,717,925	2/1998	Harper et al.	395/613	Article by Kiuchi et al., entitled: "C-HTTP—The Development of a Secure, Closed HTTP-based Network on the Internet" Published by IEEE Proceedings of SNDSS on Feb. 22, 1996, pp. 64-75.			
5,721,914	2/1998	DeVries	395/615	Article by S. Cobb, entitled: "Establishing Firewall Policy" Published by National Computer Security Assn. on Jun. 25-27, 1996, pp. 198-205.			
5,729,735	3/1998	Meyering	395/610				
5,745,360	4/1998	Leone et al.	364/140				
5,752,246	5/1998	Rogers et al.	707/10				
5,757,916	5/1998	MacDoran et al.	380/25				
5,758,150	5/1998	Bell et al.	395/610				
5,758,354	5/1998	Huang et al.	707/201				
5,758,355	5/1998	Buchanan	707/201				
5,765,171	6/1998	Gehani et al.	707/203				
5,778,346	7/1998	Frid-Nielsen et al.	395/208				

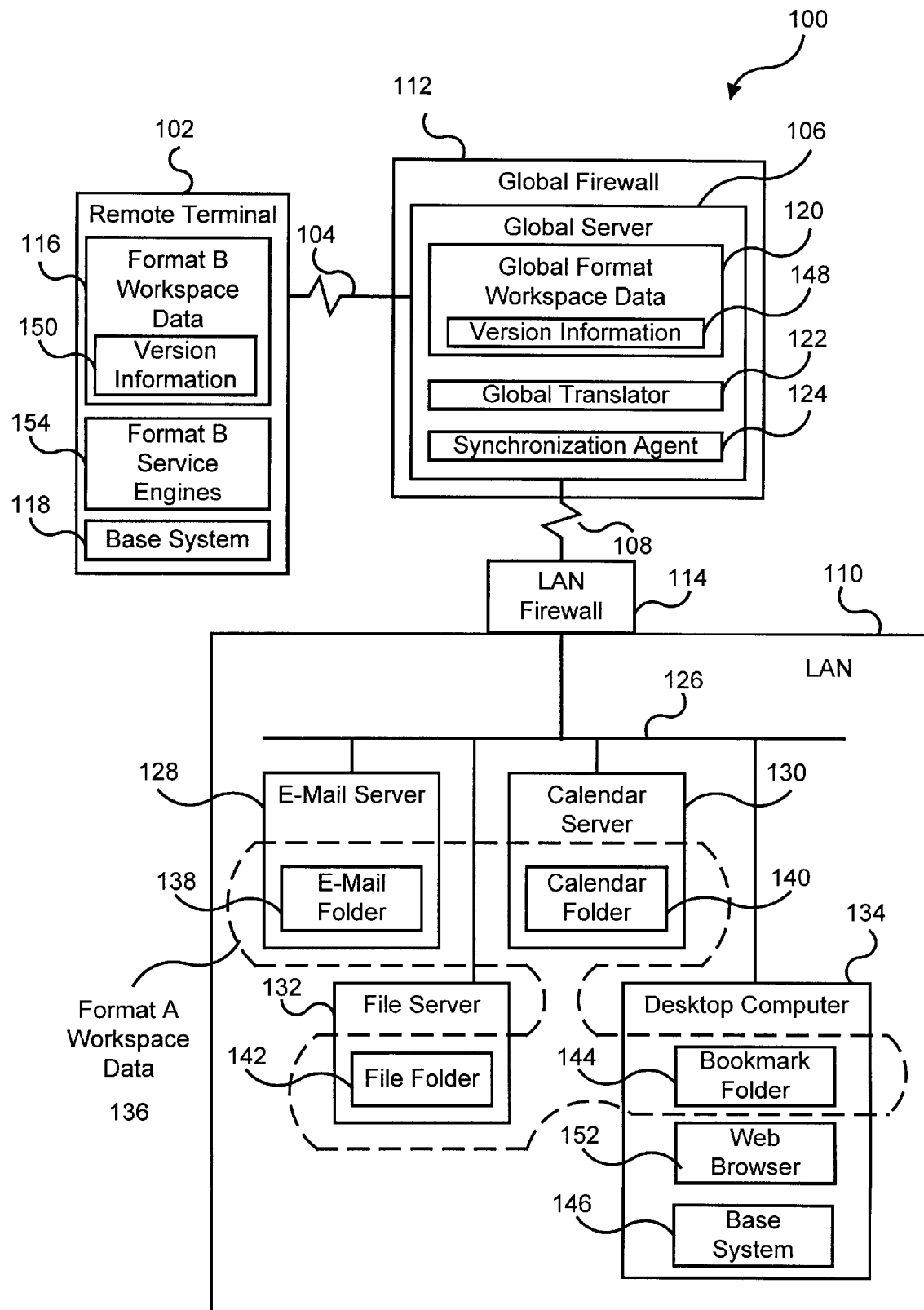


FIG. 1

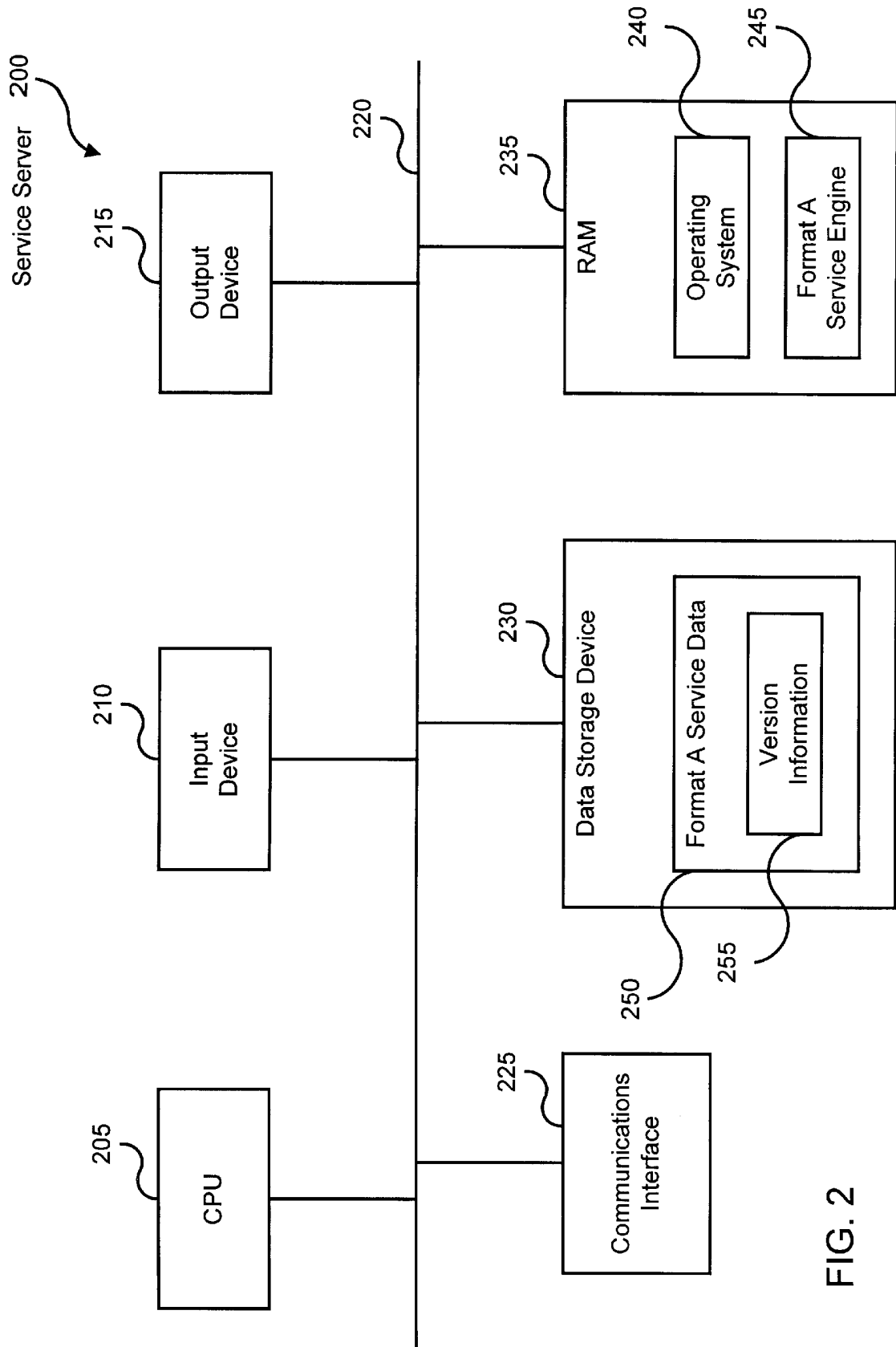


FIG. 2

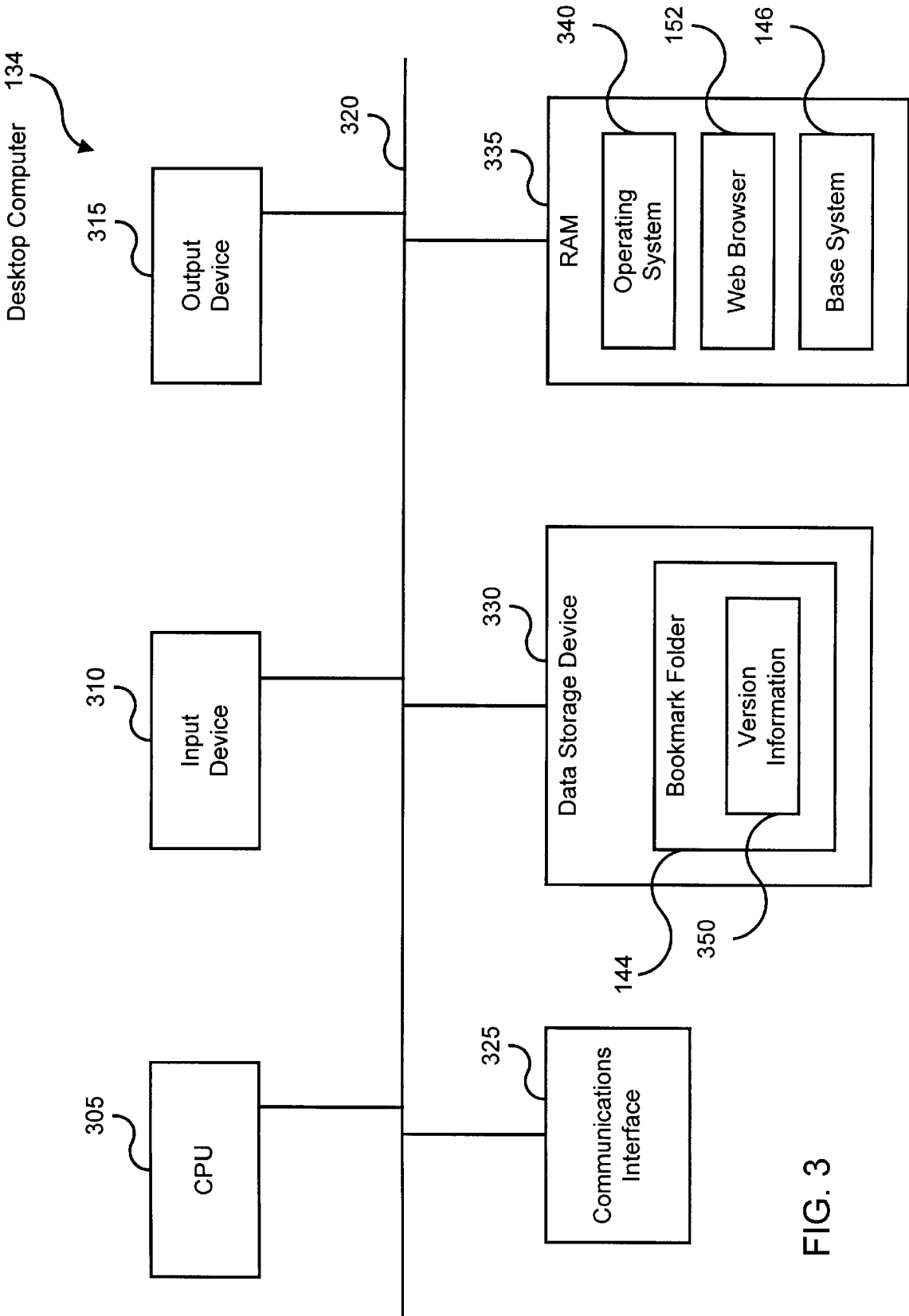


FIG. 3

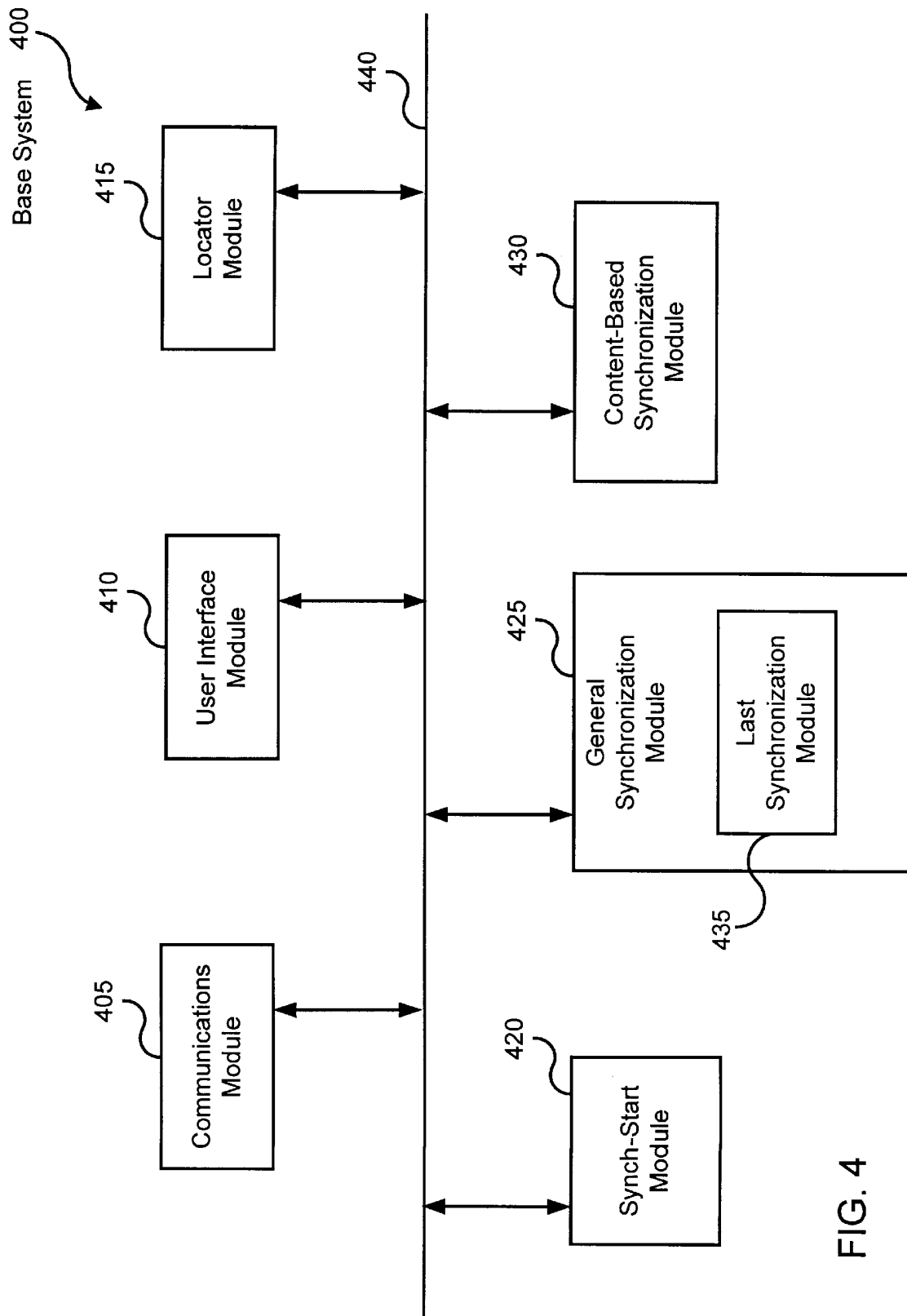


FIG. 4

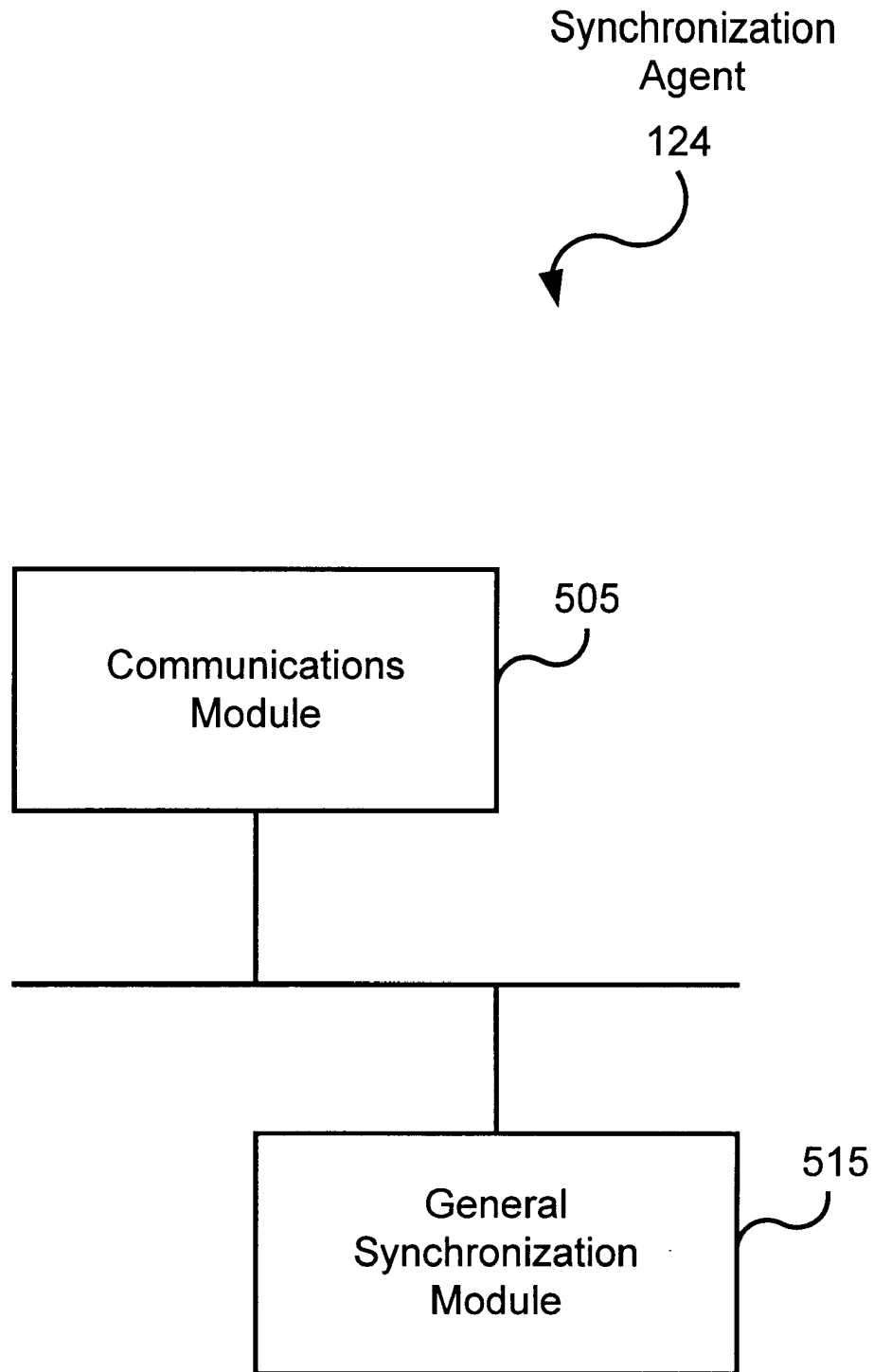


FIG. 5

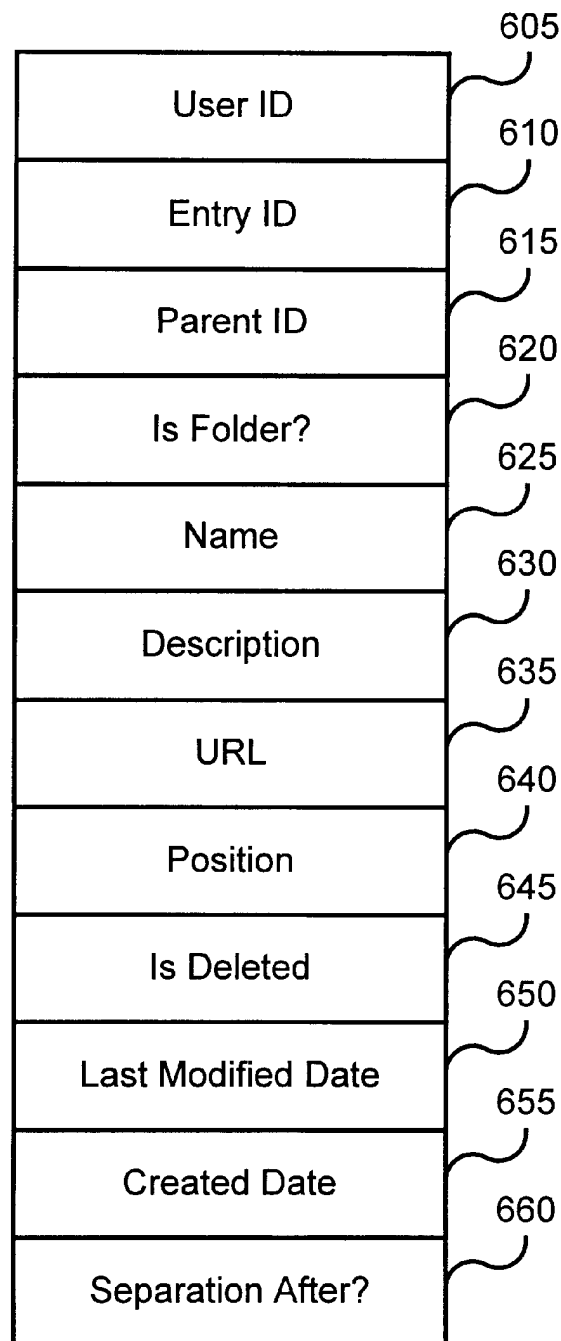
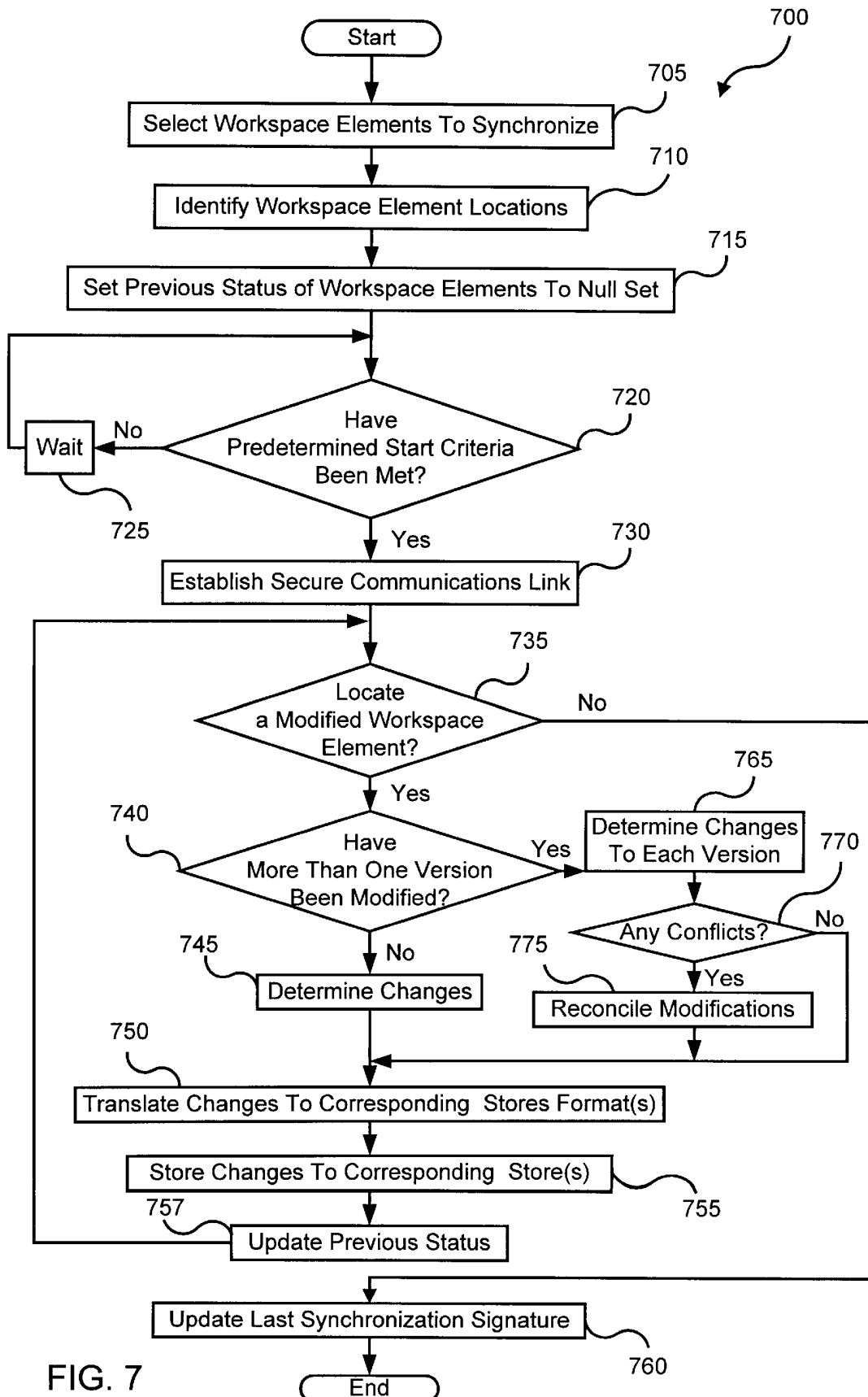


FIG. 6
(Global Format Bookmark
Example)



6,023,708

1

SYSTEM AND METHOD FOR USING A GLOBAL TRANSLATOR TO SYNCHRONIZE WORKSPACE ELEMENTS ACROSS A NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to and hereby incorporates by reference the co-pending patent application entitled "System and Method for Securely Synchronizing Multiple Copies of a Workspace Element in a Network," Ser. No. 08/835,997, filed on Apr. 11, 1997, by inventors Daniel J. Mendez, Mark D. Riggins, Prasad Wagle and Christine C. Ying. This related application has also been assigned to RoamPage, Inc.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly to a system and method for using a global translator to synchronize workspace elements such as files across a computer network.

2. Description of the Background Art

Data consistency is a significant concern for computer users. For example, when maintaining multiple independently-modifiable copies of a document, a user risks using an outdated version. By the time the user notices an inconsistency, interparty miscommunication or data loss may have already resulted. The user must then spend more time attempting to reconcile the inconsistent versions and addressing any miscommunications.

The problem of data inconsistency is exacerbated when multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.

Data consistency problems may also arise when using application programs from different vendors. For example, the Netscape Navigator™ web browser and the Internet Explorer™ web browser each store bookmarks for quick reference to interesting web sites. However, since each web browser uses different formats and stores bookmarks in different files, the bookmarks are not interchangeable. In addition, one web browser may store a needed bookmark, and the other may not. A user who, for example, runs the Internet Explorer™ web browser at home and runs the Netscape Navigator™ web browser at work risks having inconsistent bookmarks at each location.

Therefore, a system and method are needed for providing users with data consistency, and more particularly for synchronizing multiple copies of a workspace element such as a document across a computer network.

SUMMARY OF THE INVENTION

The present invention provides a system and method for using a global translator to synchronize multiple copies of a workspace element in a secure network environment. The secure network environment includes a global server connected to multiple clients. Using the present system and method, the clients automatically synchronize workspace elements between multiple sites, independent of whether the sites are protected by site firewalls. Using the present system and method, the clients can automatically synchronize work-

2

space elements across different formats and can merge workspace element folders for cross use.

The system includes a first store for storing first workspace elements in a first format, a second store for storing second workspace elements in a second format, a communications channel coupling the first store to the second store, synchronization means for synchronizing first workspace elements and second workspace elements, and a translator for translating between the first format and the second format.

Similarly, the method includes the steps of accessing a first store storing a first workspace element in a first format, accessing a second store storing a second workspace element in a second format, synchronizing the first workspace element and the second workspace element, and translating between the first format and the second format.

The system and method advantageously use a trusted third party to enable the synchronization of workspace data among multiple sites. Accordingly, a client user who maintains a work site, a home site, an off-site and the global server site can synchronize the workspace data or portions thereof among all four sites. Further, the predetermined criteria (which control when the synchronization-start module initiates synchronization) may be set so that the general synchronization module synchronizes the workspace data upon user request, at predetermined times during the day such as while the user is commuting, or after a predetermined user action such as user log-off or user log-on. Because the system and method operate over the Internet, synchronization can occur over any distance. Since the system and method include format translation, merging of workspace elements between different application programs and different platforms is possible. Further, because synchronization is initiated from within the firewall, the typical firewall which prevents inbound communications does not act as an impediment to workspace element synchronization. Also, since the user's preferences may be previously set, the present system and method may operate unattended by the client user.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a computer network in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of a FIG. 1 service server;

FIG. 3 is a block diagram illustrating details of the FIG. 1 desktop computer;

FIG. 4 is a block diagram illustrating details of a FIG. 1 base system;

FIG. 5 is a block diagram illustrating details of the FIG. 1 synchronization agent;

FIG. 6 is a graphical representation of an example bookmark in the global format; and

FIG. 7 is a flowchart illustrating a method for synchronizing multiple copies of a workspace element in a secure network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating a computer network 100, comprising a first node such as a remote computer terminal 102 coupled via a communications channel 104 such as the Internet to a global server 106. The global server 106 is in turn coupled via a communications channel 108 such as the Internet to a second node such as a Local Area

6,023,708

3

Network (LAN) **110**. The global server **106** is protected by a global firewall **112**, and the LAN **110** is protected by a LAN firewall **114**.

The LAN **110** includes a system bus **126** coupling the LAN firewall **114** to an e-mail server **128** having an e-mail folder **138** containing e-mails, to a file server **132** having a file folder **142** containing files, to a calendar server **130** having a calendar folder **140** containing calendar data, and to a desktop computer **134** having a web browser **152** and a bookmark folder **144** containing bookmarks. It will be appreciated that the e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** or portions thereof may be stored at different locations such as on the desktop computer **134**. The e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** are exemplary, grouped by like information and are collectively referred to herein as "workspace data" **136**. Those skilled in the art will recognize that the workspace data **136** may include other types of data such as an application program such as Microsoft Word 6.0.1 TM word processor and the documents created using them. It will be further appreciated that the e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** may each be divided into workspace elements, wherein each workspace element folder or each workspace element individually is identified by particular version information **255** (described below with reference to FIG. 2). Accordingly, each e-mail or e-mail folder, file or file folder, calendar or calendar folder, bookmark or bookmark folder, document or document folder, etc. may be referred to as "a workspace element."

Each workspace element of workspace data **136** in LAN **110** is maintained in a predetermined format, referred to as Format A, which is based on the service engine **245** (FIG. 2) that created it. For example, the web browser **152** on the desktop computer **134** may be the Netscape Navigator™ web browser, and the bookmarks in the bookmark folder **144** created thereby are maintained in Format A. Although Format A is being described as a single format, one skilled in the art knows that Format A actually includes a format for each information type, e.g., there will be a Format A for bookmarks, a Format A for files, a Format A for calendar data, a Format A for e-mails, etc.

The remote terminal **102** stores service engines **154** for maintaining workspace data **116**, which may include information common with information in the workspace data **136**. The workspace data **116** is maintained in a format, referred to as Format B, which may be different from Format A. Format B is also based on the service engines **154** that create the workspace elements. For example, if one of the service engines **154** is the Internet Explorer™ web browser (not shown), then the bookmarks (not shown) created therewith are maintained in Format B. Although Format B is being described as a single format, one skilled in the art knows that Format B actually includes a format for each information type. Further, the workspace data **116** also includes version information **150** similar to version information **255** described below with reference to FIG. 2.

It will be appreciated that remote terminal **102** may include a smart telephone, a Personal Data Assistant (PDA) such as the PalmPilot TMPDA the U.S. Robotics, Inc., a laptop computer, etc. As a smart telephone, the workspace data **116** may include telephone numbers and e-mails. As a PDA, the workspace data **116** may include addresses, calendar data and e-mails. As a laptop computer, the workspace data **116** may include the same types of information as workspace data **136**.

The global server **106** acts as a third party administrator. The global server **106** stores independently-modifiable cop-

4

ies of selected portions of the workspace data **136** and **116**, collectively referred to herein as workspace data **120**. Accordingly, the workspace data **120** includes an independently-modifiable copy of each workspace element in the selected portions of the workspace data **136** and **116** and an independently-modifiable copy of each corresponding version information **255** (FIG. 2) and **150**. The version information copies are collectively referred to herein as version information **148**, and are also described with reference to FIG. 2.

The global server **106** maintains the workspace data **120** in a format, referred to as a "global format," which is selected to be easily translatable by the global translator **122** to and from Format A and to and from Format B. Although the global format is being described as a single format, one skilled in the art knows that the global format actually includes a global format for each information type, e.g., there will be a global format for bookmarks, a global format for files, a global format for calendar data, a global format for e-mails, etc. An example bookmark workspace element in the global format is described in detail below with reference to FIG. 6.

Network **100** further comprises synchronization means, which includes a base system **146** stored within the LAN **110** and for example on the desktop computer **134**. Network **100** further includes a synchronization agent **124** stored outside the LAN firewall **114** and preferably on the global server **106**. The base system **146** and the synchronization agent **124** cooperate to synchronize selected portions of the workspace data **136** with selected portions of the workspace data **120**. The synchronization means may synchronize workspace elements individually, e.g., specific word processor documents, or may synchronize workspace element folders, e.g., a bookmark folder. Generally, the base system **146** manages the selected portion of the workspace data **136** within the LAN **110** and the synchronization agent **124** manages the selected portions of the workspace data **120** within the global server **106**. It will be appreciated that the global translator **122** cooperates with the synchronization means to translate data formats to and from the global format. As described in greater detail below with reference to FIG. 4, the base system **190** preferably initiates and controls data synchronization. Other components and functions of the global server **106** are described in the cross-referenced patent application which is herein incorporated by reference.

The synchronization means may also include, stored on the remote terminal **102**, a base system **118** which operates in a similar manner to the base system **146**. The base system **118** on the remote terminal **102** cooperates with the synchronization agent **124** to synchronize selected portions of the workspace data **116** with selected portions of the workspace data **120**. As described in greater detail below with reference to FIG. 4, the base system **118** on the remote terminal **102** also preferably initiates and controls data synchronization with the global server **106**. Also, note that the distribution of labor between the base system **118** in the remote terminal **102** and the synchronization agent **124** in the global server **106** may vary. Sometimes, primarily when the remote terminal **102** is a relatively less computationally powerful device (such as a smart phone or a PDA), most of the actual computationally intensive work will occur within the synchronization agent **124** in the global server **106**. In other situations, for example, when the remote terminal **102** is a fully configured PC, most of the computationally intensive work will occur locally on the base system **118** in the remote terminal **102**.

6,023,708

5

Accordingly, the synchronization means independently synchronizes the selected portions of workspace data 116 and 136 with the selected portions of the workspace data 120. Thus, the synchronization means indirectly synchronizes workspace data 136 with workspace data 116.

FIG. 2 is a block diagram illustrating details of a service server 200, wherein each of the e-mail server 145, the file server 150, the calendar server 155 and the desktop computer 160 is an instance thereof. Service server 200 includes a Central Processing Unit (CPU) 205 such as an Intel Pentium® microprocessor or a Motorola Power PC® micro-processor. An input device 210 such as a keyboard and mouse and an output device 215 such as a Cathode Ray Tube (CRT) display are coupled via a signal bus 220 to CPU 205. A communications interface 225 (such as an Ethernet port), a data storage device 230 (such as a magnetic disk), and Random-Access Memory (RAM) 235 are further coupled via signal bus 220 to the CPU 205.

An operating system 240 includes a program for controlling processing by the CPU 205, and is typically stored in the data storage device 230 and loaded into the RAM 235 for execution. A service engine 245 includes a program for performing a particular service such as maintaining an e-mail data base, a file data base, a calendar data base or a bookmarks data base. The service engine 245 may also be stored in the data storage device 230 and loaded into the RAM 235 for execution.

To perform a service, the service engine 245 creates service data 250 (e.g., an e-mail or an e-mail folder 138 containing e-mails, a file or a file folder 142 containing files, calendar data or a calendar folder 140 containing calendar data, a bookmark or a bookmark folder 144 containing bookmarks, etc.) in Format A according to predetermined protocols. The service engine 245 stores the data 250 in the data storage device 250. The service data 250 includes version information 255 indicating the date and time of the last modification and the status as of the last interaction with the global server 106.

For example, if service data 250 is created and selected to be merged with global server workspace data 120, then the version information 255 for the service data 250 may include the date of last modification and a null set indicating the status as of the last interaction with the global server 106. From the version information 255, the base system 146 determines that the service data 250 in its entirety has not been merged with the global server workspace data 120. Similarly, if the service data 255 included elements 1, 2 and 3 as of the last modification, then the previous status as of the last interaction will indicate that the service data 255 included elements 1, 2 and 3. If the service data 255 currently includes elements 2, 3 and 4, then the base system 140 will determine that, since last synchronization, element 1 has been deleted and element 4 has been added.

It will be appreciated that the version information 148 on the global server 106 includes information similar to version information 255. That is, the version information 148 will include information indicating the date and time the version was last modified and the status as of the last interaction with each client. The service engine 245 operates to update the version information 255 after modifications are made and after synchronization occurs.

FIG. 3 is a block diagram illustrating details of the desktop computer 160, which includes a CPU 305, an input device 310, an output device 315, a communications interface 325, a data storage device 330 and RAM 335, each coupled to a signal bus 320.

6

An operating system 340 includes a program for controlling processing by the CPU 305, and is typically stored in the data storage device 330 and loaded into the RAM 335 for execution. A web browser 152 (i.e., a particular service engine 245, FIG. 2) includes a Format A service program for managing bookmark folder 144 (i.e., particular service data 250, FIG. 2) which includes, version information 350 (i.e., particular version information 255, FIG. 2). The web browser 152 may be also stored in the data storage device 330 and loaded into the RAM 335 for execution. The bookmark folder 144 may be stored in the data storage device 330. As stated above with reference to FIG. 1, the base system 146 operates to synchronize the workspace data 136 (which includes the bookmark folder 144) with the workspace data 120. The base system 146 may be also stored in the data storage device 330 and loaded into the 5 RAM 335 for execution.

FIG. 4 is a block diagram illustrating details of the base system 400, which exemplifies base systems 146 and 118. Base system 400 includes a communications module 405, a user interface module 410, locator modules 415, a synchronization-start ("synch-start") module 420, a general synchronization module 425 and a content-based synchronization module 430. For simplicity, each module is illustrated as communicating with one another via a signal bus 440.

The communications module 405 includes routines for compressing data and routines for communicating via the communications interface 325 (FIG. 3) with the synchronization agent 124 (FIG. 1). The communications module 405 may further include routines for applying Secure Socket Layer (SSL) technology and user identification and authentication techniques (i.e., digital certificates) to establish a secure communication channel through the global firewall 112. Examples of communications modules 405 may include TCP/IP stacks or the AppleTalk™ protocol.

The user interface module 410 includes routines for communicating with a user, and may include a conventional Graphical User Interface (GUI). The user interface module 410 cooperates with the other system components as described herein.

The locator modules 415 include routines for identifying the memory locations of the workspace elements in the workspace data 136 or 116 and in the workspace data 120. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue boxes to query a user. More particularly, the locator modules 415 in the base system 146 determine the memory addresses of the e-mail folder 138, the file folder 142, the calendar folder 140 and the bookmark folder 144 and the memory addresses of the workspace elements therein. The locator modules 415 also determine the corresponding memory addresses of the corresponding folders in the workspace data 120 and the corresponding workspace elements therein. Similarly, the locator modules 415 in the base system 118 determine the memory locations of the workspace elements of workspace data 116 and the memory locations of the corresponding workspace elements in the workspace data 120.

It will be appreciated that the locator modules 415 may include locator modules 415 specifically dedicated to each folder or workspace data type. That is, the locator modules 415 may include a locator module 415 dedicated to locating bookmarks, a locator module 415 dedicated to locating e-mails, a locator module 415 dedicated to locating files, a locator module 415 dedicated to locating calendar

6,023,708

7

appointments, etc. It will be further appreciated that the locator modules **415** may perform workspace element memory location identification upon system boot-up or after each communication with the global server **120** to maintain updated memory addresses of workspace elements.

The synchronization-start module **420** includes routines for determining when to initiate synchronization of workspace data **136** or **116** with workspace data **120**. For example, the synchronization-start module **420** may initiate data synchronization upon user request, at a particular time of day, after a predetermined time period passes, after a predetermined number of changes, after a user action such as user log-off or upon like criteria. The synchronization-start module **420** initiates data synchronization by instructing the general synchronization module **425** (described below) to begin execution of its routines. It will be appreciated that communication with the synchronization agent **124** preferably initiates from within the LAN **10**, because the typical firewall **114** prevents in-bound communications and allows out-bound communications.

The general synchronization module **425** includes routines for receiving version information **148** for modified versions from the synchronization agent **124** (FIG. 1), and routines for examining the version information **255** or **150** against a last synchronization signature **435** (such as a last synchronization date and time) to determine which versions have been modified. The general synchronization module **425** further includes routines for examining the version information **148** and the version information **255** or **150** to determine if one or both versions of a particular workspace element or workspace element folder have been modified.

Further, the general synchronization module **425** includes routines for performing an appropriate synchronizing responsive action. Appropriate synchronizing responsive actions may include, if only one version of a workspace element in workspace data **136** or **116** has been modified, then forwarding the modified version (as the preferred version) to the other store(s) or determining and forwarding only the changes made. Computing the changes made may be performed by examining the current status against the previous status as of the last synchronization or by comparing the two versions. It will be appreciated that no content-based review of the changes is needed. It will be appreciated that one store preferably forwards only the changes to the other store for optimizing use of processor power and minimizing the data communications across the communications channel **108** or **104**.

Other appropriate synchronizing responsive actions may include, if two versions of a workspace element have been modified independently, then instructing the content-based synchronization module **430** (described below) to execute its routines. That is, if two versions of the same workspace element have been modified independently, then a content-based review of the changes is preferable. Upon completion of the data synchronization, the general synchronization module **425** updates the last synchronization signature **435**.

The content-based synchronization module **430** includes routines for reconciling two or more modified versions of a workspace element. For example, if a user has independently modified the original and the copy of a workspace element since the last synchronization, then the content-based synchronization module **430** determines an appropriate responsive action. The content-based synchronization module **430** may request the user to select a preferred one of the modified versions or may respond based on preset preferences, i.e., by storing both versions in both stores or

8

preferably by integrating the modified versions into a single preferred version which replaces each modified version at both stores.

The content-based synchronization module **430** examines the changes made to each version and determines if conflicts exist. When implementing version integration, a conflict may arise if inconsistent modifications such as deleting a paragraph in one version and modifying the same paragraph in the other version have been made. If a conflict exists, then the content-based synchronization module **430** attempts to reconcile the conflict, e.g., by requesting user selection or by storing both versions at both stores. Otherwise, if no conflict exists, then the content-based synchronization module **430** integrates the changes to each of the versions and updates the version information **148**, **150** or **255** accordingly.

FIG. 5 is a block diagram illustrating details of the synchronization agent **124**, which includes a communications module **505** (similar to the communications module **405** described above with reference to FIG. 4) and a general synchronization module **515** (similar to the general synchronization module **425** described above also with reference to FIG. 4).

The communications module **505** includes routines for compressing data, and routines for communicating via the communications channel **108** with the base system **146** or via the communications channel **104** with the base system **118**. The communications module **505** may further include routines for establishing a secure communications channel through the global firewall **112** and through the LAN firewall **114** with the communications module **405**.

Similar to the general synchronization module **425**, the general synchronization module **515** includes routines for examining the version information **148** and the last synchronization signature **435** (FIG. 4) to determine which versions have been modified and the changes made. It will be appreciated that the general synchronization module **515** may maintain its own last synchronization signature **435** copy (not shown) or may request the last synchronization signature **435** from the base system **146** or **118**. The general synchronization module **515** further includes routines for forwarding workspace data **120** determined to be modified to the general synchronization module **425**, and routines for receiving preferred versions of workspace elements of workspace data **136** or **116** or just the changes from the general synchronization module **425**.

FIG. 6 illustrates an example bookmark workspace element in the global format. The global translator **122** incorporates all the information needed by both formats (Format A and Format B) to create the Global Format. For example, if a bookmark in Format A needs elements X, Y and Z and a bookmark in Format B needs elements W, X and Y, the global translator **122** incorporates elements W, X, Y and Z to create a bookmark in the Global Format. Further, the global translator **122** incorporates the information which is needed by the synchronization means such as the last modified date. Accordingly, a bookmark in the Global Format includes a user identification (ID) **605**, an entry ID **610**, a parent ID **615**, a folder ID flag **620**, a name **625**, a description **630**, the Uniform Resource Locator (URL) **635**, the position **640**, a deleted ID flag **645**, a last modified date **650**, a created date **655** and a separation ID flag **660**.

FIG. 7 is a flowchart illustrating a method **700** for using a global translator **122** to synchronize multiple copies of a workspace element in a secure network **100**. Method **700** begins with the user interface module **410** in step **705** enabling a user to select workspace elements of workspace

6,023,708

9

data 136 and 118 for the synchronization means to synchronize. The locator modules 415 in step 710 identify the memory locations of the workspace elements in workspace data 136 and 116 and the corresponding memory locations in workspace data 120. If a selected workspace element does not have a corresponding memory location, such as in the case of adding a new workspace elements to the global server 106, then one is selected. The selected memory location may be a preexisting workspace element or a new workspace element. As stated above, workspace element memory location identification may be implemented using intelligent software or dialogue boxes. The general synchronization module 425 and general synchronization module 515 in step 715 set the previous status of the workspace elements equal to the null set. Setting the previous status to the null set indicates that all information of the workspace element has been added.

The synchronization-start module 420 in step 720 determines whether predetermined criteria have been met which indicate that synchronization of the workspace elements selected in step 705 should start. If not, then the synchronization-start module 420 in step 725 waits and loops back to step 720. Otherwise, the communications module 405 and communications module 505 in step 730 establish a secure communications channel therebetween.

The general synchronization module 425 and the general synchronization module 515 in step 735 determine whether any workspace elements have been modified. That is, the general synchronization module 425 in step 740 examines the version information 255 or 150 of each selected workspace element in the workspace data 136 or 116 against the last synchronization signature 435 to locate modified workspace elements. This comparison may include comparing the date of last modification with the date of last synchronization, or may include a comparison between the current status and the previous status as of the last interaction. Similarly, the general synchronization module 515 examines the version information 148 of each corresponding workspace element in workspace data 120 and the last synchronization signature 435 to locate modified workspace elements.

If in step 735 no modified workspace elements or folders are located, then the general synchronization modules 425 and 515 in step 760 update the last synchronization signature. 435 and method 700 ends. Otherwise, the general synchronization module 425 in step 740 determines whether more than one version of a workspace element has been modified since the last synchronization.

If only one version has been modified, then the corresponding general synchronization module 425 or 515 in step 745 determines the changes made. As stated above, determining the changes made may be implemented by comparing the current status of the workspace element against the previous status of the workspace element as of the last interaction therebetween. If the changes were made only to the version in the workspace data 120, then the global translator 122 in step 750 translates the changes to the format used by the other store, and the general synchronization module 515 in step 755 forwards the translated changes to the general synchronization module 425 for updating the outdated workspace element in the workspace data 136 or 116. If the updated version is a workspace element in the workspace data 136 or 116, then the general synchronization module 425 sends the changes to the updated version to the global translator 122 for translation and then to the general synchronization module 515 for updating the outdated workspace element in the workspace

10

data 120. The general synchronization module 425 and the general synchronization module 515 in step 757 update the previous state of to reflect the current state as of this interaction. Method 700 then returns to step 735.

If the general synchronization module 425 in step 740 determines that multiple versions have been modified, then the general synchronization module 425 in step 765 computes the changes to each version and in step 770 instructs the content-based synchronization module 430 to examine content to determine if any conflicts exist. For example, the content-based synchronization module 430 may determine that a conflict exists if a user deletes a paragraph in one version and modifies the same paragraph in another version. The content-based synchronization module 430 may determine that a conflict does not exist if a user deletes different paragraphs in each version. If no conflict is found, then method 700 jumps to step 750 for translating and forwarding the changes in each version to the other store. However, if a conflict is found, then the content-based synchronization module 430 in step 775 reconciles the modified versions. As stated above, reconciliation may include requesting instructions from the user or based on preselected preferences performing responsive actions such as storing both versions at both stores. Method 700 then proceeds to step 750.

It will be appreciated that in step 710 new workspace elements and preexisting workspace elements to which new workspace elements will be merged are set to "modified" and the previous status is set to the null set. Thus, the general synchronization module 425 in step 740 will determine that more that one version has been modified and the content-based synchronization module 430 in step 770 will determine that no conflict exists. The changes in each will be translated and forwarded to the other store. Accordingly, the two versions will be effectively merged and stored at each store.

For example, if a first bookmark folder was created by the web browser 152 on the desktop computer 134, a second folder was created by a web browser (not shown) on the remote terminal 102, no preexisting folder existed on the global server 106 and the user selected each of these folders for synchronization, then the synchronization means will effectively merge the first and second folders. That is, the general synchronization module 425 on the desktop computer 134 will determine that the first folder has been modified and the previous status is equal to the null set. The general synchronization module 425 will determine and send the changes, i.e., all the workspace elements in the first folder, to a new global folder on the global server 106. Similarly, the general synchronization module 425 on the remote terminal 102 will determine that, as of its last interaction, the previous status of each of the second and the global folders is the null set. The general synchronization module 425 will instruct the content-based synchronization module 430 to examine the changes made to each folder to determine whether a conflict exists. Since no conflicts will exist, the general synchronization module 425 will forward the changes to the global folder and the general synchronization module 515 will forward its changes to the second store, thereby merging the workspace elements of the first and second folders in the global and second folders. The general synchronization module 515 will inform the general synchronization module 425 that the global folder has been modified relative to the last interaction, and will forward the new changes to the first folder. Thus, the first and second folders will be merged and stored at each store.

For a second example, the user may select an exemplary document in the LAN 110 to be synchronized. The general

6,023,708

11

synchronization module **425** will forward the document to the global server **106**. Similarly, the user may select the same document for synchronization on the remote terminal **102**. The general synchronization module **515** will forward the document to the remote terminal **102**. If changes were made to the documents independently, then the content-based synchronization module **430** will examine the content of the documents to determine if a conflict exists. If no conflict exists, then as described above, the general synchronization modules **425** and **515** will merge the documents. Otherwise, if a conflict does exist, the content-based synchronization module **430** will reconcile the changes and then the general synchronization modules **425** and **515** will forward the reconciled changes to each other.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the present invention. For example, although the global server **106** is illustrated as a single device, the global server **106** may include several computers networked together. Components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The system is limited only by the following claims.

What is claimed is:

1. A system, comprising:

a first store for storing a first workspace element in a first format;

a second store for storing a second workspace element which is an independently modifiable copy of the first workspace element in a second format;

a communications channel coupling the first store to the second store;

synchronization means for synchronizing the first workspace element and the second workspace element; and
a translator for translating between the first format and the second format.

2. The system of claim **1**, wherein the first format is the same as the second format.

3. The system of claim **1** wherein the second format includes HTML.

4. The system of claim **1**, wherein the first store is in a LAN.

5. The system of claim **1**, further comprising a firewall for protecting the first store.

6. The system of claim **1**, further comprising a firewall for protecting the second store.

7. The system of claim **1**, wherein the synchronization means includes

a first general synchronization module for examining the first workspace element at the first store to determine whether it has been modified; and

a second general synchronization module for examining the second workspace element at the second store to determine whether it has been modified.

8. The system of claim **7**, wherein

the first workspace element and the second workspace element each include version information, and

the first and second general synchronization modules each examine the version information of each workspace

12

element to a last synchronization signature to determine whether the workspace element was modified.

9. The system of claim **1**, wherein the synchronization means includes a synchronization-start module for determining when to initiate synchronization.

10. The system of claim **1**, wherein the first workspace element includes a first folder and the second workspace element includes a second folder.

11. The system of claim **10**, further comprising a content-based synchronization module for examining content of the first workspace element and of the second workspace element when the first workspace element and the second workspace element have both been modified since the last synchronization.

12. The system of claim **11**, wherein the content-based synchronization module determines if any conflicts exist, and if so then the content-based synchronization module reconciles the conflicts.

13. The system of claim **11**, wherein the content-based synchronization module determines if any conflicts exist, and if not then the content-based synchronization module integrates the changes made to each workspace element.

14. The system of claim **1**, further comprising a third store coupled to the second store for storing a third workspace element in a third format.

15. The system of claim **14**, wherein the translator translates between the second format and the third format.

16. The system of claim **8**, wherein the version information indicates the status of each workspace element as of the last synchronization.

17. A method, comprising:

accessing a first store storing a first workspace element in a first format;

accessing a second store storing a second workspace element which is an independently modifiable copy of the first workspace element in a second format;

synchronizing the first workspace element and the second workspace element; and

translating between the first format and the second format.

18. The method of claim **17**, wherein the first format is the same as the second format.

19. The method of claim **17**, wherein the second format includes HTML.

20. The method of claim **17**, wherein the first store is in a LAN.

21. The method of claim **17**, wherein the first store is protected by a firewall.

22. The method of claim **17**, wherein the second store is protected by a firewall.

23. The method of claim **17**, further comprising examining the first workspace element to determine whether it has been modified; and examining the second workspace element to determine whether it has been modified.

24. The method of claim **23**, wherein the first workspace element and the second workspace element each include version information; and

further comprising the step of comparing the version information of each workspace element to a last synchronization signature to determine whether the workspace element was modified.

25. The method of claim **17**, further comprising the step of determining when to initiate synchronization.

26. The method of claim **17**, wherein the first workspace element includes a first folder and the second workspace element includes a second folder.

6,023,708

13

27. The method of claim 26, further comprising the step of examining content of the first workspace element and of the second workspace element when the first workspace element and the second workspace element have both been modified since the last synchronization.

28. The method of claim 27, further comprising the step of determining if any conflicts exist, and if so then reconciling the conflicts.

29. The method of claim 27, further comprising the step of determining if any conflicts exist, and if not then integrating the changes made to each workspace element.

30. The method of claim 17, further comprising the step of accessing a third store coupled to the second store storing, a third workspace element in a third format.

31. The method of claim 30, wherein further comprising the step of translating between the second format and the third format.

32. The method of claim 24, wherein the version information indicates the status of each workspace element as of the last synchronization.

33. A system, comprising:

means for accessing a first store storing a first workspace element in a first format;

14

means for accessing a second store storing a second workspace element which is an independently-modifiable copy of the first workspace element in a second format;

means for synchronizing the first workspace element and the second workspace element; and

means for translating between the first format and the second format.

34. A computer-readable storage medium storing program code for causing a computer to perform the steps of:

accessing a first store storing a first workspace element in a first format;

accessing a second store storing a second workspace element which is an independently modifiable copy of the first workspace element in a second format;

synchronizing the first workspace element and the second workspace element; and

translating between the first format and the second format.

* * * * *